



# *TrailsCommunity*

Rapport de projet - Assitant Offroad

BELLANGER Stéphane

MONNIER Ysée

DESHORS Yann

Novembre 2016 - Decembre 2016

# Table des matières

<b>I</b>	<b>Spécification des exigences logicielles</b>	<b>4</b>
<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Objectifs . . . . .	5
1.2	Portée . . . . .	5
1.3	Définitions . . . . .	5
1.4	Références . . . . .	6
1.5	Aperçu . . . . .	6
1.6	Répartition du travail . . . . .	6
1.7	Définitions . . . . .	7
1.7.1	Termes . . . . .	7
1.8	Références . . . . .	7
<b>2</b>	<b>Description générale</b>	<b>8</b>
2.1	Perspective du produit . . . . .	8
2.2	Fonctions du produit . . . . .	8
2.3	Classes et caractéristiques des utilisateurs . . . . .	8
2.4	Environnement d'exploitation . . . . .	9
2.5	Contraintes de conception et d'implémentation . . . . .	9
2.6	Documentation utilisateur . . . . .	9
2.7	Hypothèses et dépendances . . . . .	9
<b>3</b>	<b>Exigences de l'interface externe</b>	<b>10</b>
3.1	Interfaces des utilisateurs . . . . .	10
3.2	Interfaces logicielles . . . . .	10
3.3	Interfaces de communication . . . . .	10
<b>4</b>	<b>Caractéristiques du système</b>	<b>11</b>
4.1	Fonction système 1 . . . . .	11
4.1.1	Description et priorité . . . . .	11
4.1.2	Séquence de stimulation / réponse . . . . .	11
4.1.3	Exigences fonctionnelles . . . . .	11
4.2	System Feature 2 (and so on) . . . . .	12
<b>5</b>	<b>Autres exigences non fonctionnelles</b>	<b>13</b>
5.1	Exigences de performance . . . . .	13
5.2	Safety Requirements . . . . .	13

5.3	Exigences de sécurité . . . . .	13
5.4	Attributs de qualité du logiciel . . . . .	13
<b>6</b>	<b>Autres exigences</b>	<b>14</b>
6.1	Annexe A : Glossaire . . . . .	14
6.2	Annexe B : Modèles d'analyse . . . . .	14
6.3	Annexe C : Liste à déterminer . . . . .	14
<b>II</b>	<b>Serveur TrailsCommunity</b>	<b>15</b>
<b>1</b>	<b>Introduction</b>	<b>16</b>
1.1	Conventions . . . . .	16
1.2	Codes statut . . . . .	16
<b>2</b>	<b>Ressources</b>	<b>17</b>
2.1	Authentification . . . . .	17
2.1.1	Requête . . . . .	17
2.1.2	Réponse . . . . .	17
2.2	Inscription . . . . .	18
2.2.1	Requête . . . . .	18
2.2.2	Réponse . . . . .	19
2.3	Ajout d'une session . . . . .	19
2.3.1	Requête . . . . .	19
2.3.2	Réponse . . . . .	20
2.4	Liste des sessions . . . . .	20
2.4.1	Requête . . . . .	20
2.4.2	Réponse . . . . .	21
2.5	Session . . . . .	22
2.5.1	Requête . . . . .	22
2.5.2	Réponse . . . . .	23
2.6	Mis à jour d'une session . . . . .	24
2.6.1	Requête . . . . .	24
2.6.2	Réponse . . . . .	24
2.7	Rejoindre une session . . . . .	25
2.7.1	Requête . . . . .	25
2.7.2	Réponse . . . . .	25
2.8	Ajout d'un waypoint . . . . .	25
2.8.1	Requête . . . . .	25
2.8.2	Réponse . . . . .	26

## Table des figures

# Première partie

## Spécification des exigences logicielles

# Chapitre 1

## Introduction

Cette section fournit une vue d'ensemble de tout ce qui est inclus dans ce document SRS. De plus, l'objectif de ce document est décrit et une liste de définitions est fournie.

### 1.1 Objectifs

L'objectif de ce document est de fournir une description détaillée des conditions requises pour l'application TrailsCommunity. Il illustrera le but et la description complète du développement du système. Il expliquera également les contraintes du système, l'interface et les interactions avec les applications externes. Ce document est principalement destiné à être proposé à un client pour son approbation. C'est aussi une référence pour le développement de la première version du système pour l'équipe de développement.

### 1.2 Portée

TrailsCommunity est une application mobile GPS qui permet aux utilisateurs de connaître en temps réel le positionnement des autres participants d'une activité de pleine air en cours. L'ensemble des activités sont regroupés en type qui seront limités à une liste pré-définie. L'application doit être gratuite et simple d'utilisation. L'organisateur de l'activité devra fournir deux adresses. Une pour le point de départ de l'application et la deuxième pour le point d'arrivée. La création des sessions se fera directement sur l'application mobile. En outre, le logiciel a besoin à la fois d'Internet et de connexion GPS pour récupérer et afficher les résultats. Tous les systèmes d'informations sont conservés dans une base de données, qui se trouve sur un serveur web. Le logiciel interagit également avec l'API Google Map qui permettra d'afficher et de tracer les différents parcours des utilisateurs. De plus, le téléphone doit aussi comprendre un GPS pour pouvoir localiser l'utilisateur à tout moment. L'application a également la capacité de représenter à la fois des informations sommaires et détaillées des différentes statistiques de l'utilisateur courant mais aussi des autres participants de la session en cours.

### 1.3 Définitions

session    une session est une activité disponible dans l'application

visiteur personne n'étant pas identifier

utilisateur personne identifié à l'application. Il possède les choix de créer ou de rejoindre une session

organisateur utilisateur ayant créé une session et pouvant la gérer

participant utilisateur qui a rejoint une session

waypoint désigne un point de la route a atteindre ou doit avoir lieu un changement de cap

## 1.4 Références

The Institute of Electrical and Electronic Engineer NY USA IEEE Recommended Practice for SRS,  
<http://www.utdallas.edu/~chung/RE/IEEE830-1993.pdf>

Chalmers IEEE standards - SRS Example, 2010, [http://www.cse.chalmers.se/~feldt/courses/reqeng/examples/srs\\_example\\_2010\\_group2.pdf](http://www.cse.chalmers.se/~feldt/courses/reqeng/examples/srs_example_2010_group2.pdf)

Droid5 Informatics Pvt Ltd androidhive, 2016, <http://www.androidhive.info>

Oracle Java 8 Documentation, 2016, <https://docs.oracle.com/javase/8/docs/api/>

Google Android Documentation, 2016, <https://developer.android.com/guide>

James Britt Ruby Doc, 2016, <http://ruby-doc.org>

Rails Community Ruby On Rails Documentation, 2016, <http://rubyonrails.org>

## 1.5 Aperçu

//Reprendre a partir d'ici

L'objectif de ce projet est de réaliser une application Android d'assistant offroad. L'application créé devra permettre d'enregistrer, de visualiser et de partager des positions GPS entre un groupe d'utilisateur. Cette application pourra donc être utilisé dans le cadre d'activité en groupe et en plein air(randonnées, painball, vélo). De plus, chaque utilisateur pourra visualiser différentes statistiques en rapport avec ses activités réalisées précédemment. Il est plus que nécessaire pour ce genre d'application d'avoir un temps de réponse rapide du serveur. C'est pour cela que la stratégie principal du groupe est de réalisé un serveur dédié uniquement pour l'application. De plus, les exigences sur les règles de codage sont appliqué sur le client.

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

## 1.6 Répartition du travail

La majorité de la phase de conception a été réalisée à trois : les spécifications et la modélisation de l'axe fonctionnel ont été faites par le groupe au complet. Les diagrammes de séquence et d'activité ont été répartis entre nous, puis nous les avons tous passé en revue et modifié jusqu'à que nous soyons tous d'accord sur l'ensemble des diagrammes et des idées qu'elles véhiculent.

Lors de la phase de développement, nous avons réparti les rôles de la manière suivante :

- Ysée : Conception UML, développement du serveur, développement de l'application mobile
- Stéphen : Conception UML, développement de l'application mobile, rapport
- Yann : Conception UML, rapport

Toutefois, vu l'objectif du projet, nous avons tous participé au développement du client, mais à un niveau d'implication différent en fonction des compétences des membres du groupe.

## 1.7 Définitions

### 1.7.1 Termes

## 1.8 Références

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>



# Chapitre 2

## Description générale

### 2.1 Perspective du produit

Est ce que c'est pertinent ? On en parle déjà dans "Intended Audience and Reading Suggestions" et "Objectifs"

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

### 2.2 Fonctions du produit

Même chose ici. Voir -> "Project Scope Description du projet"

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>

### 2.3 Classes et caractéristiques des utilisateurs

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

## 2.4 Environnement d'exploitation

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

## 2.5 Contraintes de conception et d'implémentation

<Describe any items or issues that will limit the options available to the developers. These might include : corporate or regulatory policies ; hardware limitations (timing requirements, memory requirements) ; interfaces to other applications ; specific technologies, tools, and databases to be used ; parallel operations ; language requirements ; communications protocols ; security considerations ; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

## 2.6 Documentation utilisateur

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

## 2.7 Hypothèses et dépendances

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

# Chapitre 3

## Exigences de l'interface externe

### 3.1 Interfaces des utilisateurs

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

### 3.2 Interfaces logicielles

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

### 3.3 Interfaces de communication

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

# Chapitre 4

## Caractéristiques du système

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

### 4.1 Fonction système 1

<Don't really say "System Feature 1." State the feature name in just a few words.>

#### 4.1.1 Description et priorité

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

#### 4.1.2 Séquence de stimulation / réponse

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

#### 4.1.3 Exigences fonctionnelles

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1 : REQ-2 :

## 4.2 System Feature 2 (and so on)

# Chapitre 5

## Autres exigences non fonctionnelles

### 5.1 Exigences de performance

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

### 5.2 Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

### 5.3 Exigences de sécurité

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

### 5.4 Attributs de qualité du logiciel

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are : adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

# Chapitre 6

## Autres exigences

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

### 6.1 Annexe A : Glossaire

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

### 6.2 Annexe B : Modèles d'analyse

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

### 6.3 Annexe C : Liste à déterminer

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>

# Deuxième partie

## Serveur TrailsCommunity



# Chapitre 1

## Introduction

### 1.1 Conventions

- **Client** : Application client.
- **Statut** : Code réponse de la requête HTTP
- Toutes les réponses possibles sont listés sous la catégorie 'Réponse'
- Toutes les réponses sont du format JSON

### 1.2 Codes statut

Tous les codes de statut sont des standards HTTP. Cette API utilisera les suivants :

- *2XX* - Succès de la requête
- *4XX* - Erreur du côté client
- *5XX* - Erreur du côté serveur

Code statut	Description
200	Requête traitée avec succès
201	Requête traitée avec succès et création d'un document
202	Requête traitée avec succès mais pas d'information à renvoyer
400	La syntaxe de la requête est erronée
401	Une authentification est nécessaire pour accéder à la ressource
404	Ressource non trouvée
500	Erreur interne du serveur
503	Service temporairement indisponible ou en maintenance

# Chapitre 2

## Ressources

### 2.1 Authentification

#### 2.1.1 Requête

Méthode	URL
<i>POST</i>	api/login/auth_token

Type	Paramètres	Valeurs
<i>HEADER</i>	Application/json	<i>String</i>
<i>JSON</i>	{ "auth": { "email": "mail@mail.com", "password": "FZ,f235nDE", } }	<i>String</i> <i>String</i>

*email* : correspond à l'adresse mail valide de l'utilisateur

*password* : mot de passe de l'utilisateur

#### 2.1.2 Réponse

Statut	Réponse
200	{ "jwt": <Authorization> }
401	
500	

*Authorization(string)* - Tous les autres appels nécessitant une authentification doivent avoir cette clé dans le header.

## 2.2 Inscription

### 2.2.1 Requête

Enregistrement d'un utilisateur dans le système.

Méthode	URL
<i>POST</i>	api/users/

Type	Paramètres	Valeurs
<i>HEADER</i>	Application/json	<i>String</i>
<i>JSON</i>	{ "nickname": "Arthur83", "email": "apaul@custom.com", "phone_number": "+33651678908", "password": "myPassword33", "password_confirmation": "myPassword33" }	<i>String</i> <i>String</i> <i>String</i> <i>String</i> <i>String</i>

*nickname* : pseudonyme de l'utilisateur

*email* : email de l'utilisateur non nul et valide

*password* : mot de passe d'utilisateur

*password\_confirmation* : confirmation du mot de passe de l'utilisateur. Il doit être identique à *password*.

## 2.2.2 Réponse

Statut	Réponse
200	<pre>{   "data": {     "id": 2,     "nickname": "Arthur83",     "phone_number": "+33651678908",     "current_session_id": null   } }</pre>
400	<pre>{   "errors": {     "nickname": [       "has already been taken"     ],     "email": [       "is invalid"     ]   } }</pre>
500	

## 2.3 Ajout d'une session

### 2.3.1 Requête

Permet d'ajouter une nouvelle session dans le serveur. Cette ressource est contrainte d'un token d'authentification *<Authorization>* qui a été donnée avec la ressource *api/login/auth\_token*

Méthode	URL
<i>POST</i>	<i>api/sessions/</i>

Type	Paramètres	Valeurs
<i>HEADER</i>	<i>Application/json</i>	<i>String</i>
<i>HEADER</i>	<i>Authorization</i>	<i>String</i>
<i>JSON</i>	<pre>{   "activity": 1,   "password": "lockSessionPsw",   "departure_place": "43.179363;5.717782",   "arrival_place": "43.191168;5.730819",   "start_date": "2016-11-20" }</pre>	<i>Integer</i> <i>String</i> <i>String</i> <i>String</i> <i>Date</i>

*activity* : identifiant de l'activité. (1 : randonnée, 2 : vélo, ...)  
*password* : mot de passe de la session, permet de restreindre l'entrée.  
*departure\_place* : lieu de départ. Couple de position géographique (*latitude*; *longitude*)  
*arrival\_place* : lieu d'arrivée. Couple de position géographique (*latitude*; *longitude*)

## 2.3.2 Réponse

Statut	Réponse
200	<pre>{   "data": {     "id": 5,     "activity": 1,     "departure_place": "43.179363;5.717782",     "arrival_place": "43.191168;5.730819",     "start_date": "2016-11-20",     "close": false,     "lock": true,     "user": {       "id": 1,       "nickname": "Raptor234",       "phone_number": "+33623569450",       "current_session_id": null     },     "coordinates": [],     "waypoints": []   } }</pre>
400	<pre>{   "errors": {     "activity": [       "can't be blank",       "is not a number"     ]   } }</pre>
401	
500	

## 2.4 Liste des sessions

### 2.4.1 Requête

Permet de récupérer la liste des sessions du système. Cette ressource est contrainte d'un token d'authentification *<Authorization>* qui a été donnée avec la ressource *api/login/auth\_token*

Méthode	URL
<i>GET</i>	api/sessions/

Type	Paramètres	Valeurs
<i>HEADER</i>	Authorization	<i>String</i>

## 2.4.2 Réponse

Statut	Réponse
200	<pre>{   "data": [     {       "id": 1,       "activity": 1,       "departure_place": "43.179363;5.717782",       "arrival_place": "43.191168;5.730819",       "start_date": "2016-11-20",       "close": false,       "lock": true,       "user": {         "id": 2,         "nickname": "Arthur83",         "phone_number": "+33651678908",         "current_session_id": null       }     },     {       "id": 2,       "activity": 3,       "departure_place": "43.179363;5.717782",       "arrival_place": "43.191168;5.730819",       "start_date": "2016-11-20",       "close": false,       "lock": true,       "user": {         "id": 2,         "nickname": "Arthur83",         "phone_number": "+33651678908",         "current_session_id": null       }     }   ] }</pre>
401	
500	

## 2.5 Session

### 2.5.1 Requête

Permet de récupérer une session spécifique selon son identifiant (ID). Cette ressource est contrainte d'un token d'authentification *<Authorization>* qui a été donnée avec la ressource *api/login/auth\_token*

Méthode	URL
<i>GET</i>	<code>api/sessions/:id</code>

Type	Paramètres	Valeurs
<i>URL_PARAM</i>	<code>id</code>	<i>Integer</i>
<i>HEADER</i>	<code>Authorization</code>	<i>String</i>

*id* : identifiant de la session voulu

## 2.5.2 Réponse

Statut	Réponse
200	<pre>{   "data": {     "id": 2,     "activity": 3,     "departure_place": "43.179363;5.717782",     "arrival_place": "43.191168;5.730819",     "start_date": "2016-11-20",     "close": false,     "lock": true,     "user": {       "id": 2,       "nickname": "Arthur83",       "phone_number": "+33651678908",       "current_session_id": null     },     "coordinates": [       {         "latitude": 43.179363,         "longitude": 5.717782,         "user_id": 1       }     ],     "waypoints": [       {         "latitude": 43.179363,         "longitude": 5.717782       }     ]   } }</pre>
401	
404	<pre>{   "errors": {     "session": "Record Not Found"   } }</pre>
500	



## 2.6 Mis à jour d'une session

### 2.6.1 Requête

Permet de mettre à jour session spécifique selon un identifiant (ID) dans le système. Cette ressource est contrainte d'un token d'authentification *<Authorization>* qui a été donnée avec la ressource *api/login/auth\_token*

Méthode	URL
<i>PUT</i>	<code>api/sessions/:id</code>

Type	Paramètres	Valeurs
<i>HEADER</i>	Application/json	<i>String</i>
<i>HEADER</i>	Authorization	<i>String</i>
<i>JSON</i>	<pre>{   "close": true }</pre>	<i>boolean</i>

*close* : boolean déterminant si la session doit être fermée.

### 2.6.2 Réponse

Statut	Réponse
200	<pre>{   "data": {     "id": 7,     "activity": 3,     "departure_place": "43.179363;5.717782",     "arrival_place": "43.191168;5.730819",     "start_date": "2016-11-20",     "close": true,     "lock": true,     "user": {       "id": 2,       "nickname": "Arthur83",       "phone_number": "+33651678908",       "current_session_id": 1     },     "coordinates": [],     "waypoints": []   } }</pre>
401	
500	

## 2.7 Rejoindre une session

### 2.7.1 Requête

Permet de à l'utilisateur de rejoindre une session spécifique selon son identifiant (ID). Cette ressource est contrainte d'un token d'authentification *<Authorization>* qui a été donnée avec la ressource *api/login/auth\_token*

Méthode	URL
<i>GET</i>	<code>api/sessions/:id/join</code>

Type	Paramètres	Valeurs
<i>URL_PARAM</i>	<code>id</code>	<i>Integer</i>
<i>HEADER</i>	<code>Authorization</code>	<i>String</i>

*id* : identifiant de la session voulu

### 2.7.2 Réponse

Statut	Réponse
200	
401	
404	<pre>{   "errors": {     "session": "Record Not Found"   } }</pre>
500	

## 2.8 Ajout d'un waypoint

### 2.8.1 Requête

Permet de à l'utilisateur d'ajouter une nouveau waypoint pour une session spécifique selon son identifiant (ID). Cette ressource est contrainte d'un token d'authentification *<Authorization>* qui a été donnée avec la ressource *api/login/auth\_token*

Méthode	URL
<i>GET</i>	<code>api/sessions/:id/waypoint</code>

Type	Paramètres	Valeurs
<i>URL_PARAM</i>	id	<i>Integer</i>
<i>HEADER</i>	Authorization	<i>String</i>
<i>JSON</i>	{ "latitude": 43.179363, "longitude": 5.717782 }	<i>Double</i> <i>Double</i>

*id* : identifiant de la session voulu

## 2.8.2 Réponse

Statut	Réponse
200	
401	
404	{ "errors": { "session": "Record Not Found" } }
500	